HEALTH INSURANCE CLAIM

A Mini Project Report

submitted by

Panchami.P.M(MES20MCA-2038)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

February 2022

DECLARATION

I undersigned hereby declare that the project report **Health Insurance Claim**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of Ms.Priya.J.D, Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:Kuttippuram .....

Date:

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **Health Insurance Claim** is a bonafide record of the Mini Project work carried out by **Panchami.P.M(MES20MCA-2038)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

At the very outset I would like to thank the almighty's mercy towards me over the years. . . I wish to express my sincere thanks to my project guide Ms.Priya J.D Assistant Professor, Dept. of Master of Computer Applications who guided me for the successful completion of this project. I also thank her for valuable suggestions, guidance, constant encouragement, boundless corporation, constructive comments and motivation extended to me for completion of this project work. I would express my sincere thanks to Mr.Hyderali K, Associate Professor and Head of Department for his immense guidance to complete the project successfully. I would like to express my sincere thanks to all the faculty members of Master of Computer Applications department for their support and valuable suggestion for doing the project work. Last but not least my graceful thanks to my parents, friends and also the persons who supported me directly and indirectly during the project.

Panchami.P.M(MES20MCA-2038)

Abstract

In this project I have developed a web application for health insurance claims. In this project a system and method for a rules-based pre-adjudication of a benefits claim submission is disclosed. The reviewing and adjudicating medical insurance claims is done electronically. In support medical procedure eligibility verification, data entry editing, electronic claim submission, claim status determination and electronic remittance processing is also done. The Health Insurance Claim helps us in creating the model for each registration and keeps tracking of claim management.This type of web application is mainly accessible to four people, namely, admin, health provider, insurance company, and users.Insurance company processes the claim and updates the status for each claim. Advantage associated with the system is quicker claims processing and remittance, reduced transaction fees and direct connections with insurance companies.Here the health provider shares some medical related documents to users, and users will also send some of the medical documents with specific required ID proofs to insurance company,so in order to secure these documents i used AES Encryption algorithm to encrypt these documents.AES is one of the secure and fast algorithm

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The main purpose of this project is to develop a web application for health insurance claims.Medicare plays a vital role in a real time environment. Medicare can contribute to a significant part of a country's economy. It provides a direct link between the user and the insurance company.This project overcomes the shortcomings of the existing system.This project is database based.Its web part was developed in Python,in Flask frame work.It contains different tables where all the data's are stored.Health insurance claims record patient health care treatments and expenses and, although created for the health care payment system, are potentially useful for research.Corresponding reports for each patient should be sent by the health provider using the id of user.Also the user can access that documents.Similarly,the user can send claim request to the corresponding insurance company.

1.1.1 Motivation

Health insurance is an agreement whereby insurance company agrees to undertake a guarantee of compensation for medical expenses in case the insured falls ill or meets with an accident which leads to hospitalization of the insured. Generally, insurance companies have tie-ups with the leading hospitals so as to provide cashless treatment to the insured. In case the insurance company has no tie-ups with the hospital, they reimburse the cost of expenses incurred by the insured. The government also promotes health insurance by providing a deduction

from income tax.There are several benefits for having an insurance policy.The main benefits are Cashless Treatment,Medical Checkup,Tax Benefit etc; Now the health insurance claim is done manually.This may cause money loss and also it is waste of time.Thus this project will overcome all of this drawbacks.

1.2 Objective

In today's world Medicare plays a vital role in a real time environment. Medicare can add to a noteworthy piece of a nations economy. Medicare is the maintenance or improvement of health. Access to health care varies across countries, groups, and individuals, largely influenced by social and economic conditions as well as the health policies in place. I am focusing on building an effective system for health insurance. Here the claiming is done electronically.Also it gives a direct connection between Insurance Companies and Users.

1.3 Report Organization

The project report is divided into four sections. Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

- Sourabh Chandra, Smita Paira,"A comparative survey of symmetric and Asymmetric key cryptography", International Conference on Electronics, Communication and Computational Engineering, 2014

A comparative survey of symmetric and asymmetric key cryptography Written by Sourabh Chandra Assistant Professor Department of Computer Science amp; Engineering, Calcutta Institute of Technology Kolkata, India. In this paper, we have surveyed the traditional algorithms, along with the proposed algorithms based on their pros and cons, related to Symmetric and Asymmetric Key Cryptography. The proposed algorithms proved to be highly efficient in their respective grounds but there are certain areas that remained open, related to these algorithms, and have not yet been thoroughly discussed. This paper also presents an appropriate future scope related to these open fields. we have highlighted the basic as well as proposed algorithms related to these cryptographic techniques.

A. Symmetric Key Cryptography Is also called secret-key or shared key cryptography. In this mechanism, the sender and receiver shares a common key for both encryption and decryption. the sharing of this key becomes sometimes insecure.

B. Asymmetric Key Cryptography Is known as public key cryptography. In this technique, the sender uses a public key of the receiver for encryption and the receiver uses his private key to decrypt the message. The concept of selfcertification is absent here instead digital signatures are used to certify the keys.

- Bawna Bhat,Abdul Wahid Ali,Apurva Gupta,“ DES and AES Performance Evaluation”, International Conference on Computing, Communication and Automation (ICCCA), 2015

Written by Bawna Bhat School of Computer Science and Engineering Galgotias University Greater Noida, UP, India In this paper discuss about the Cryptography. When we transmit a multimedia data such as audio, video, images etc. over the network, cryptography provides security. In cryptography, we encode data before sending it and decode it on receiving, for this purpose, we use many cryptographic algorithms

A. Data Encryption Standard Is a symmetric key algorithm for encrypt the data, to secure from attacker or unauthorized party. DES provides the influence in security world to protect the information.

B. Advanced Encryption Standard As DES key is small in size and its processor power has less technological advancement is Advanced Encryption Standard.

C. Cryptography and Classification Cryptography is the protecting technique of data from the unauthorized party by converting into the non-readable form. The classification of cryptography on the basis of key based algorithm, there are following two types of algorithms such as:

(i) symmetric key based algorithm, sometimes known as conventional key algorithm and

(ii) asymmetric key based algorithm, also known as public-key algorithm

- International Research Journal of Engineering and Technology (IRJET)

  PAPER TOWARDS PRIVACY ASSURED HEALTH INSURANCE CLAIMS: The medical insurance claim process is carried out by healthcare providers, insurance companies, and clearinghouses. Clearinghouses may intentionally or unintentionally leak critical health records. This paper proposes a distributed system to replace the role of middlemen during the process of claiming the health insurance and to reduce the risk of data leakage.Data structures for patient information, medical record, insurance payment, and insurance policies are designed within the ledger. The paper focused on defining smart contracts for privacy assurance, as well as automating the insurance claim process.

Chapter 3

Methodology

3.1 Introduction

Medicare plays a vital role in a real time environment. Medicare can contribute to a significant part of a country's economy. Medicare is the maintenance or improvement of health. Access to health care varies across countries, groups, and individuals, largely influenced by social and economic conditions as well as the health policies in place. In this project a system and method for a rules-based pre-adjudication of a benefits claim submission is disclosed. The system can handle both individual and family insurance policies.Currently the health insurance claim is done manually, that cause money loss and also its waste a lot of valuable time.In order to overcome this i have developed a web application for health insurance claim which done electronically.So this project contains four modules,They are Admin, Health Provider, Insurance Company and Users. The Health provider and the insurance company can register themselves in this web application. Requests from insurance companies and health providers are verified by the admin. The verified user will receive a username and password.Thus the Health provider and Insurance company can login using that credentials. The Health Provider can add User information.After the successful Registration,the user will get an email containing password and the username will be the password.The User can login using this details.The Health provider can send all the medical related documents to the User.And the User can download the documents.The Insurance company can add several policies.The policy contains Policy name,Amount,Description and Rules.The User can select policies from several policy and can initiate the claim request.Also User uploading all the related documents that corre-

sponds to the policy.The request sent by the User can view by the corresponding Insurance company.The Insurance company can verify the details and attached documents of Users.If all the requirements are met, the insurance company can grant the claim.The main advantage of the system is, User can view the status of their claim.Also user can file any complaint that related to policies to the admin.The admin can view the complaints and will give reply to each complaints.

In this project in order to secure the documents of the Users, I used AES encryption technique.Advanced Encryption Standard (AES) is a specification for the encryption of electronic data. AES is a block cipher.The key size can be 128/192/256 bits.Encrypts data in blocks of 128 bits each.

Here the files are uploaded to 'request.file'. After this, saving this file to a temporary file in a folder. Here a folder will be created to store the encrypted files. Also creating a buffer size of 64 * 1024bytes file. AES algorithm is based on a key. Calling pyAesencrypt function, giving the corresponding path of encryption, also specifying key and buffer size. Now the file is encrypted, the corresponding file will be in encpath,it cannot be accessible. Also its removing the temporary file. Now its empty. The key and path of encrypted file is inserted at the time of insertion.

At the time of decryption, get the corresponding name of file. The buffer size will be same, and key is also used here. So the document will be stored in the encrypted folder.Then using pyAesdecrypt file, the encrypted file will be decrypted to a specific decrypt path. Here the send file is giving for downloading. So we can access the encrypted file.

MODULES

1.Admin

a.Health provider management.

b.Insurance company management.

c.Claim related complaint management.

2.Health Provider

a.Account creation

b.Add and view User's information.

c.Add Health care reports of User.

3.Insurance Company

a.Registration.

b.Claim policy management.

c.Claim request management.

d.Health provider phr verification related to claim.

e.Claim User document verification

f.Sanctioning of claim.

4.Users

a.Policy Selection.

b.Initiate claim request.

c.Document uploads.

d.View claim status.

These are the main functions in the modules. Other functions are, Here health provider and insurance company management is the management of their joining requests to admin. The health provider can add users information and view the added informations.They can edit the information of users. The insurance company can add different policies. In addition, they manage the requests from the users.They are verifying all the documents from users.They can grant a claim based on the evidence. This project is based on database.So the Backend is Mysql.And the Front end is Python(Flask).

Agile Methodology

The project was developed using Agile Development Model.The entire project was divided into four sprints.In the first sprint, the registration of the insurance company and the health provider was done.In the second sprint, claim policy management, policy selection, and initiate claim request was done.In third sprint add view user's information, claim request management, health provider phr verification, also adding healthcare reports of users was done.And in the last sprint insurance company claim related complaint management was done also document upload ,verification, sanctioning of claim and view claim status was done.

Product Backlog

A product backlog is a prioritized list of work.It contains all the functions in a project.Here I divided my project into 4 sprints.I divided my project user stories based on priority.Based on the difficulty and size of the user stories in the product backlog, I divided it into two, high and medium priorities.

User Story ID	Priority <High/Medium/Low>	Size (Hours)	Sprint <#>	Status <Planned/In progress/Completed>	Release Date	Release Goal
1	High	4	1	Completed	05/12/2021	Health provider registration
4	Medium	2		Completed	18/12/2021	Account creation
7	High	4		Completed	25/12/2021	Registration

Figure 3.1: Product Backlog

8	High	4	2	Completed	30/12/2021	Claim policy management
13	Medium	2		Completed	09/01/2022	login
14	Medium	3		Completed	18/01/2022	Policy Selection
15	High	4		Completed	28/01/2022	Initiate claim request

Figure 3.2: Product Backlog

5	**Medium**	**3**	**3**	**Completed**	**30/01/2022**	**Add and view users information**
6	Medium	3		Completed	31/01/2022	Add Health care reports of users(for claim sanctioning purpose
9	Medium	2		Completed	02/02/2022	Claim request management
10	High	4		Completed	04/02/2022	Health provider phr verification related to claim

Figure 3.3: Product Backlog

2	**Medium**	**2**	**4**	**Completed**	**06/02/2022**	**Insurance company Management**
3	Medium	2		Completed	08/02/2022	Claim related complaint management
11	High	4		Completed	12/02/2022	Claim users document verification
12	Medium	3		Completed	15/02/2022	Sanctioning of claim
16	Medium	2		Completed	17/02/2022	Document uploads
17	Medium	2		Completed	19/02/2022	View claim status

Figure 3.4: Product Backlog

User Stories

User stories are short, simple descriptions of a feature told from the perspective of the person.Here I have divided my project into 4 modules.They are admin,user,insurance company and health provider.The project contains 17 user stories.

User Story ID	As a <type of user>	I want to	So that I can
1	Admin	Health Provider Management	Health provider registration
2	Admin	Insurance company Management	Verification of health insurance company
3	Admin	Claim related complaint management	Dealing of complaint related to user claim and sent reply
4	Health provider	Account creation	Account creation request sent to admin, admin verifies the request and sent confirmation via email
5	Health provider	Add and information	Management of users
6	Health provider	Add Health care reports of users (for claim sanctioning purpose)	Claim records uploads with detailed narration
7	Insurance company	Registration	Account creation
8	Insurance company	Claim policy management	Registration of different policies and its specification management
9	Insurance company	Claim request management	Claim request management
10	Insurance company	Health provider phr verification related to claim	Document verification
11	Insurance company	Claim users document verification	Verification of users uploaded document with health provider report
12	Insurance company	Sanctioning of claim	Claim sanctioning/ rejection/ adjustment of claim
13	Users	Login	Login
14	Users	Policy Selection	View policy and select policy
15	Users	Initiate claim request	Create claim request
16	Users	Document uploads	Upload various documents related to claim
17	Users	View claim status	View status of request

Figure 3.5: User Stories

Project Plan

Project planning is a discipline addressing how to complete a project in a certain time frame.The project is divided into 4 sprints. Each sprint contains certain functions .The approximate time between the start date and the end date is calculated depending on the function.Here also indicates the status of the project.

User story ID	Task name	Start date	End date	Hours	Status
1	Sprint1	01/12/2021	05/12/2021	10	Completed
4		08/12/2021	18/12/2021		Completed
7		22/12/2021	25/12/2021		Completed

Figure 3.6: Project Plan

8	Sprint 2	28/12/2021	30/12/2021	13	Completed
13		05/01/2022	09/01/2022		Completed
14		12/01/2022	18/01/2022		Completed
15		22/01/2022	28/01/2022		Completed

Figure 3.7: Project Plan

User story ID	Task name	Start date	End date	Hours	Status
Sprint3					
5	Sprint 3	29/01/2022	30/01/2022	12	Completed
6		30/01/2022	31/01/2022		Completed
9		03/02/2022	02/02/2022		Completed
10		12/02/2022	04/02/2022		Completed

Figure 3.8: Project Plan

Sprint4					
2	Sprint 4	05/02/2022	06/02/2022	15	Completed
3		07/02/2022	08/02/2022		Completed
11		09/02/2022	12/02/2022		Completed
12		13/02/2022	15/02/2022		Completed
16		16/02/2022	17/02/2022		Completed
17		18/02/2022	19/02/2022		Completed

Figure 3.9: Project Plan

Sprint Backlog Plans

During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story.It also specifies the time it takes for user stories to be completed in less than 14 days for four sprints.

Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story #1,4,7		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Health provider registration	05/12/2021	3	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Account creation	18/12/2021	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Registration	25/12/2021	5	1	0	2	1	0	0	0	1	0	0	0	0	0	0
Total		10	3	2	3	1	0	0	0	1	0	0	0	0	0	0

Figure 3.10: Sprint Backlog-Plan-Sprint1

Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story #8,13,14,15		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Claim policy management	30/12/2021	4	1	1	1	0	0	1	0	0	0	0	0	0	0	0
Account creation	09/01/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Policy Selection	18/01/2022	3	1	0	0	1	0	0	0	1	0	0	0	0	0	0
Initiate claim request	28/01/2022	4	0	1	1	0	1	1	0	0	0	0	0	0	0	0
Total		13	3	3	2	1	1	2	0	1	0	0	0	0	0	0

Figure 3.11: Sprint Backlog-Plan-Sprint2

Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story #5,6,9,10		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Add Users information	30/12/2021	3	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Add Health care reports of users	31/12/2021	3	1	1	0	1	0	0	0	0	0	0	0	0	0	0
Claim request management	02/01/2022	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Health provider phr verification related to claim	04/01/2022	4	1	0	2	1	0	0	0	0	0	0	0	0	0	0
Total		12	4	2	3	3	0	0	0	0	0	0	0	0	0	0

Figure 3.12: Sprint Backlog-Plan-Sprint3

Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story #2,3,11,12,16,17		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Insurance company Management	06/02/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Claim related complaint management	08/02/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Claim Users document verification	12/02/2022	4	1	0	2	1	0	0	0	0	0	0	0	0	0	0
Sanctioning of claim	15/02/2022	3	0	0	0	1	0	0	1	0	1	0	0	0	0	0
Document uploads	17/02/2022	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0
View claim status	19/02/2022	2	0	0	1	0	0	0	0	1	0	0	0	0	0	0
Total		15	4	2	3	3	0	0	1	1	1	0	0	0	0	0

Figure 3.13: Sprint Backlog-Plan-Sprint4

Sprint Backlog Actual

Similiar to Sprint Backlog plan.It specifies the actual time it takes for user stories to be completed in less than 14 days for four sprints.

Backlog Item	Status & completion date	Original estimate in hours	Day1 2/3	Day2 3/3	Day3 4/3	Day4 5/3	Day5 6/3	Day6 7/3	Day7 8/3	Day8 9/3	Day9 10/3	Day10 11/3	Day11 12/3	Day12 13/3	Day13 14/3	Day14 15/3	Completed(Y/N)
User story #1,4,7		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	
Health provider Registration	05/12/2021	3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	Y
Account Creation	18/12/2021	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Y
Insurance company registration	25/12/2021	5	1	0	2	1	0	0	0	1	0	0	0	0	0	0	Y
Total		10	3	2	3	1	0	0	0	1	0	0	0	0	0	0	

Figure 3.14: Sprint Backlog-Actual-Sprint1

Backlog Item	Status & completion date	Original estimate in hours	Day1 2/3	Day2 3/3	Day3 4/3	Day4 5/3	Day5 6/3	Day6 7/3	Day7 8/3	Day8 9/3	Day9 10/3	Day10 11/3	Day11 12/3	Day12 13/3	Day13 14/3	Day14 15/3	Completed(Y/N)
User story #8,13,14,15		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	
Claim policy management	30/12/2021	4	1	1	1	0	0	1	0	0	0	0	0	0	0	0	Y
Account creation	09/01/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Y
Policy Selection	18/01/2022	3	1	0	0	1	0	0	0	1	0	0	0	0	0	0	Y
Initiate claim request	28/01/2022	4	0	1	1	0	1	1	0	0	0	0	0	0	0	0	Y
Total		13	3	3	2	1	1	2	0	1	0	0	0	0	0	0	

Figure 3.15: Sprint Backlog-Actual-Sprint2

Backlog Item	Status & completion date	Original estimate in hours	Day1 2/3	Day2 3/3	Day3 4/3	Day4 5/3	Day5 6/3	Day6 7/3	Day7 8/3	Day8 9/3	Day9 10/3	Day10 11/3	Day11 12/3	Day12 13/3	Day13 14/3	Day14 15/3	Completed(Y/N)
User story #5,6,9,10		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	
Add users information	30/12/2021	3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	Y
Add Health care reports of user	31/12/2021	3	1	1	0	1	0	0	0	0	0	0	0	0	0	0	Y
Claim request management	02/01/2022	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	Y
Health provider phr verification related to claim	04/01/2022	4	1	0	2	1	0	0	0	0	0	0	0	0	0	0	Y
Total		12	4	2	3	3	0	0	0	0	0	0	0	0	0	0	

Figure 3.16: Sprint Backlog-Actual-Sprint3

Backlog Item	Status & completion date	Original estimate in hours	Day1 2/3	Day2 3/3	Day3 4/3	Day4 5/3	Day5 6/3	Day6 7/3	Day7 8/3	Day8 9/3	Day9 10/3	Day10 11/3	Day11 12/3	Day12 13/3	Day13 14/3	Day14 15/3	Completed(Y/N)
User story #2,3,11,12,16,17		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Insurance company Management	06/02/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Y
Claim related complaint management	08/02/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Y
Claim users document verification	12/02/2022	4	1	0	2	1	0	0	0	0	0	0	0	0	0	0	Y
Sanctioning of claim	15/02/2022	3	0	0	0	1	0	0	1	0	1	0	0	0	0	0	Y
Document uploads	17/02/2022	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	Y
View claim status	19/02/2022	2	0	0	1	0	0	0	0	1	0	0	0	0	0	0	Y
Total		15	4	2	3	3	0	0	1	1	1	0	0	0	0	0	

Figure 3.17: Sprint Backlog-Actual-Sprint4

3.2 Results

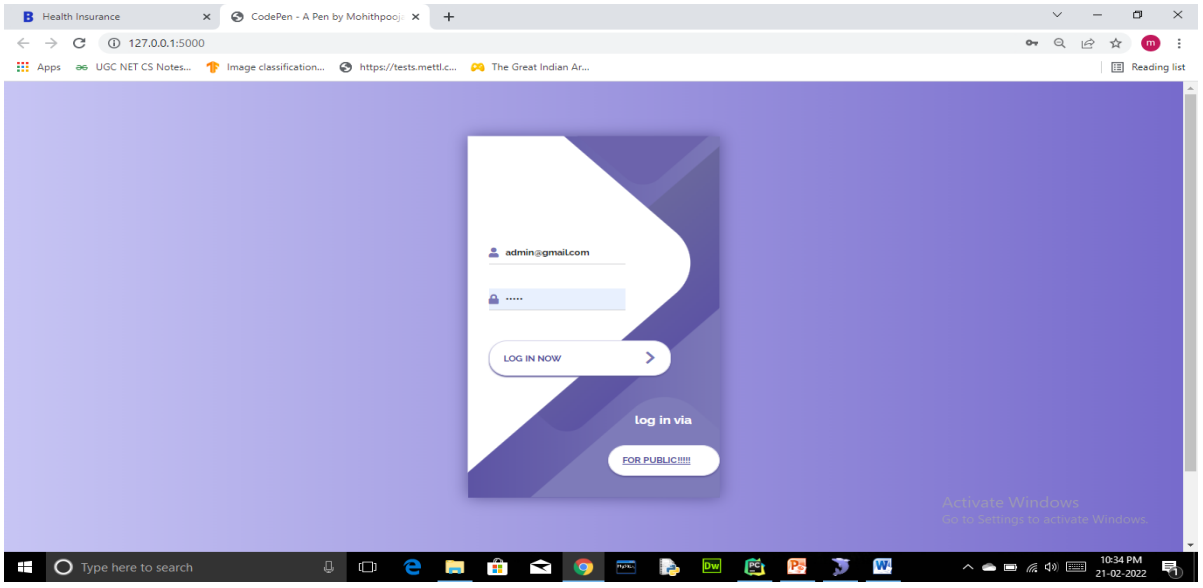

Figure 3.18: Output-Login Page

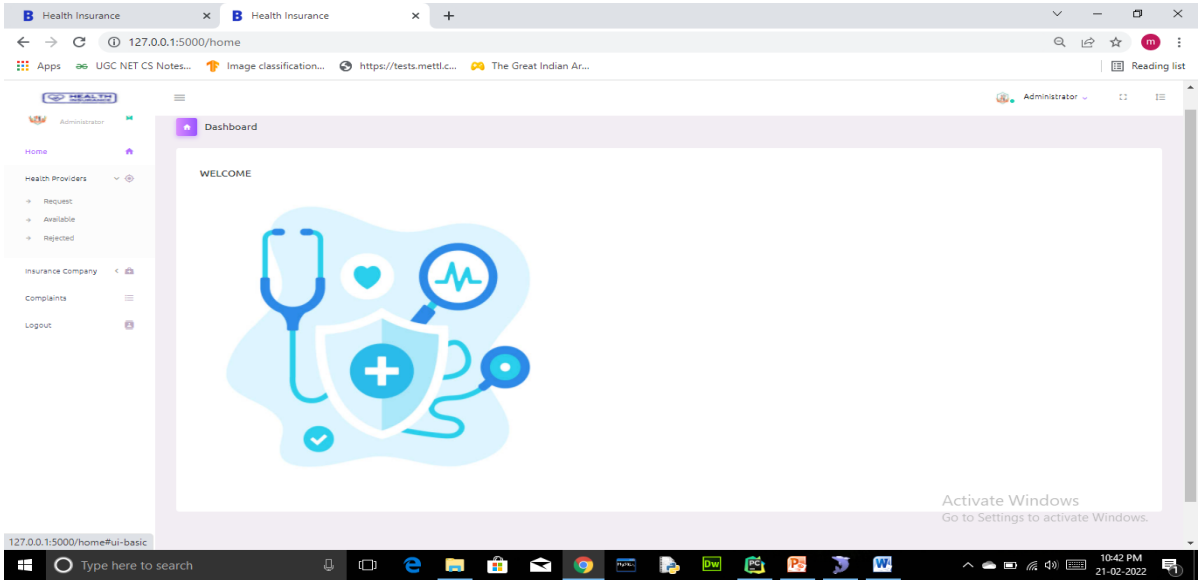

Figure 3.19: Output-Health home page

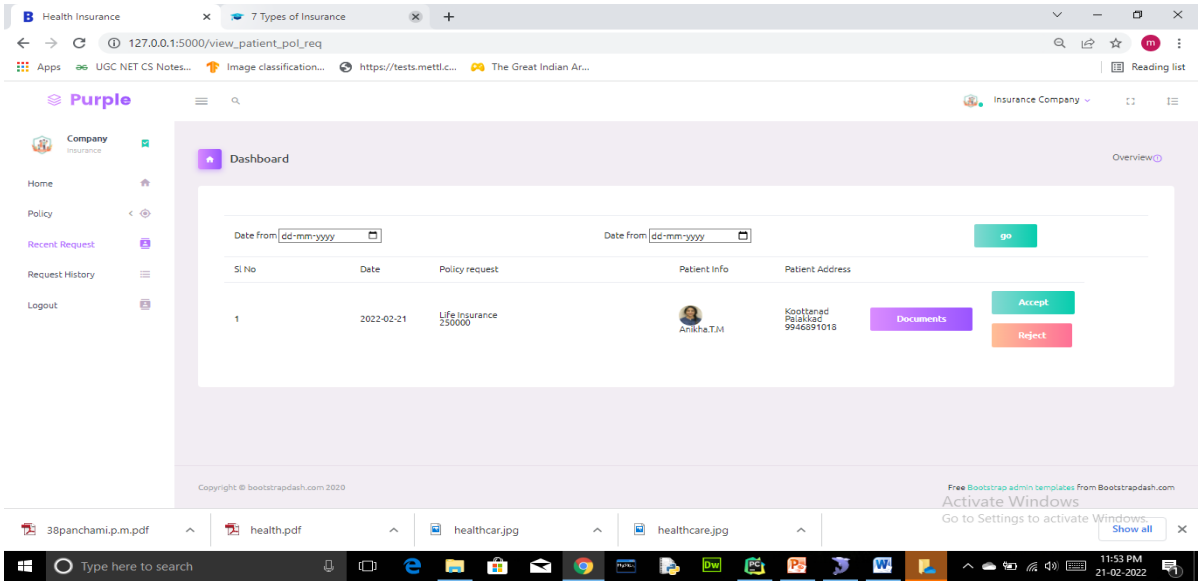

Figure 3.20: Output-View

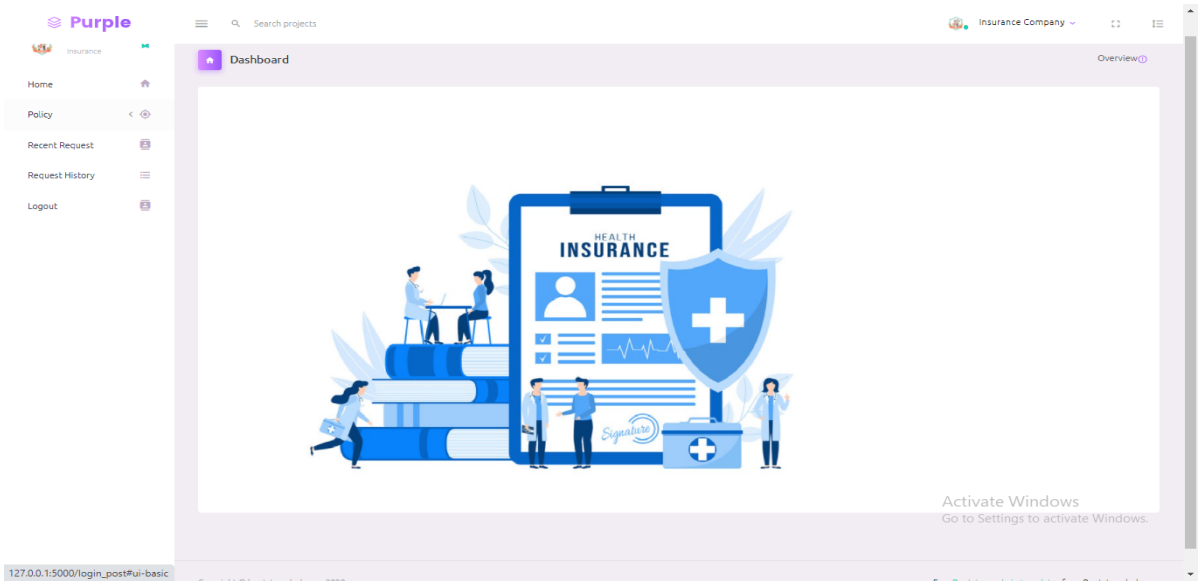

Figure 3.21: Output-Insurance company home

Chapter 4

Conclusions

In the past couple of years, there has been a high demand for insurance applications. Since 2017, providers and agents have shown some interest in web apps for the insurance company. People started trusting web applications that helped them make better decisions regarding insurance.The best insurance web app was the one that initially introduced people to comparing different policies on their platform. It enabled them to understand the price of each policy, terms, regulations, maturity amount, and a lot of other details. One of the biggest insurance app advantages was that people could now pay for insurance online. So I developed a insurance web application named Health Insurance Claim.Also the main advantage of this system is that it does not waste much time and money So this Health Insurance web application helps you to maintain a direct connection with the Insurance company.This will enable the users to apply for various policies and also to know the status of the claim.

References

[1] **Sourabh Chandra, Smita Paira** Paira,"A comparative survey of symmetric and Asymmetric key cryptography", International Conference on Electronics, Communication and Computational Engineering, 2014

[2] **KBawna Bhat,Abdul Wahid Ali,Apurva Gupta** DES and AES Performance Evaluation", International Conference on Computing, Communication and Automation (ICCCA), 2015

[3] International Research Journal of Engineering and Technology (IRJET)

[4] www.w3schools.com/html
www.codeproject.com
www.stackoverflow.com

Appendix

Source Code

```
import os
import random
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

import pyAesCrypt
from flask import Flask, render_template, request, session, send_file
from flask_mail import Mail,Message


from DBConnection import Db
app = Flask(__name__)

app.secret_key="hi"
app.config['MAIL_SERVER'] = 'smtp.gmail.com'
app.config['MAIL_PORT'] = 587
app.config['MAIL_USE_TLS'] = True
app.config['MAIL_USERNAME'] = ''
app.config['MAIL_DEFAULT_SENDER'] = 'mediclaim2022@gmail.com'
app.config['MAIL_PASSWORD'] = '9946891018'
mail = Mail(app)

staticpath="C:\\Users\\Admin\\PycharmProjects\\flaskProject\\static\\"
@app.route('/')
def login():
    return render_template("ll.html")

@app.route('/login_post',methods=['post'])
def login_post():
    name=request.form['textfield']
    password=request.form['textfield2']

    qry="select * from login where username='"+name+"' and password='"+password+"'"
    print(qry)
    db=Db()
    res=db.selectOne(qry)
    if res!=None:
        type=res['type']
        session["lid"]=res["lid"]
        if type=='admin':
            return home()
```

```
        elif type=='health':
            return render_template("health/home.html")
        elif type=='patient':
            return render_template("patients/patient_home.html")
        elif type=='insurance':
            return render_template("/insurance/insurance_home.html")
        else:
            return 'invalid'
    else:
        return '''<script>alert("user not allowed"); window.location='/'</script>'''

@app.route('/ic_registration')
def ic_registration():
    return render_template("/insurance/ic_registration.html")

@app.route('/ic_registration',methods=['post'])
def ic_registration_post():
    name=request.form['textfield']
    about = request.form['textarea']
    place=request.form['textfield2']
    post=request.form['textfield3']
    pin=request.form['textfield5']
    district=request.form['textfield4']
    state=request.form['textfield6']
    number=request.form['textfield7']
    email=request.form['textfield8']
    website=request.form['textfield9']

    db=Db()
    qry="INSERT INTO
          insurance_company(name,about,place,post,pin,district,state,email,number,website,status)value('"+name+"','"+about+"','"+place+"','"+pos
    print(qry)
    res=db.insert(qry)
    return '''<script>alert("Registered"); window.location='/'</script>'''
@app.route('/viewpending_ic')
def viewpending_ic():
    qry="select * from insurance_company where status='pending'"
    ob=Db()
    res=ob.select(qry)
    return render_template("insurance.html",data=res)
@app.route('/icrequest_accept/<id>')
def icrequest_accept(id):
    ob = Db()
    qry="update insurance_company set status='Approved' where icid='"+str(id)+"'"
    res=ob.update(qry)
    qry1 = "select * from insurance_company where icid='" + str(id) + "'"
    res1 = ob.selectOne(qry1)
    email=res1['email']
    number = res1['number']
    qry2 = "insert into login(username,password,type)values('" + str(res1['email']) + "','" +str(res1['number']) +
          "','insurance')"
    res2 = ob.insert(qry2)

    qq="update insurance_company set iclid='"+str(res2)+"' where icid='"+str(id)+"'"
    res3=ob.update(qq)
    # msg = Message(subject="------------------------",
    #           sender=app.config.get("mediclaim2022@gmail.com"),
    #           recipients=[email],
    #           body="Username : " + email + " & " + "Password : " + str(number))
    # mail.send(msg)

    s = smtplib.SMTP(host='smtp.gmail.com', port=587)
    s.starttls()
```

Appendix

```
    s.login("mediclaim2022@gmail.com", "9946891018")
    msg = MIMEMultipart() # create a message........."
    message = "Messege from mediclaim"
    msg['From'] = "mediclaim2022@gmail.com"
    msg['To'] = email
    msg['Subject'] = "Your Password for mediclaim"
    body = "Your Account has been verified by our team. You Can login using your password - " + str(str(number)+"Your
        username="+str(email))
    msg.attach(MIMEText(body, 'plain'))
    s.send_message(msg)

    return '''<script>alert("accepted"); window.location='/viewpending_ic'</script>'''

@app.route('/icreq_reject/<id>')
def icreq_reject(id):
    qry = "update insurance_company set status='Rejected' where icid='" + str(id) + "'"
    ob = Db()
    res = ob.update(qry)
    return '''<script>alert("rejected"); window.location='/viewpending_ic'</script>'''


@app.route('/signup')
def signup():
    return render_template("/health/signup hp.html")

@app.route('/signup_post',methods=['post'])
def signup_post():
    name=request.form['textfield']
    about=request.form['textarea']
    place = request.form['textfield2']
    post=request.form['textfield3']
    pin = request.form['textfield4']
    district = request.form['textfield5']
    state = request.form['textfield6']
    number = request.form['textfield7']
    email = request.form['textfield8']
    website = request.form['textfield9']
    db=Db()

    qry="insert into
        health_provider(hp_name,about,place,post,pin,district,state,number,email,website,status)values('"+name+"','"+about+"','"+place+"','"+p
    res=db.insert(qry)
    return '''<script>alert("Registered"); window.location='/'</script>'''

@app.route("/home")
def home():
    return render_template("home.html")



@app.route('/viewhealthpending_request')
def viewhealthpending_request():
    qry="select * from health_provider where status='pending'"
    ob=Db()
    res=ob.select(qry)
    return render_template("view healthp.html",data=res)

@app.route('/healthproviderrequest_accept/<id>')
def healthproviderrequest_accept(id):
    ob = Db()
    qry="update health_provider set status='Approved' where hpid='"+str(id)+"'"
    res=ob.update(qry)
    qry1 = "select * from health_provider where hpid='" + str(id) + "'"
```

```
    res1 = ob.selectOne(qry1)
    email=res1['email']
    number = res1['number']
    qry2 = "insert into login(username,password,type)values('" + str(res1['email']) + "','" +str(res1['number']) +
          "','health')"
    res2 = ob.insert(qry2)

    qq="update health_provider set hplid='"+str(res2)+"' where hpid='"+str(id)+"'"
    res3=ob.update(qq)
    # msg = Message(subject="------------------------",
    #            sender=app.config.get("mediclaim2022@gmail.com"),
    #            recipients=[email],
    #            body="Username : " + email + " & " + "Password : " + str(number))
    # mail.send(msg)

    s = smtplib.SMTP(host='smtp.gmail.com', port=587)
    s.starttls()
    s.login("mediclaim2022@gmail.com", "9946891018")
    msg = MIMEMultipart() # create a message........."
    message = "Messege from mediclaim"
    msg['From'] = "mediclaim2022@gmail.com"
    msg['To'] = email
    msg['Subject'] = "Your Password for mediclaim"
    body = "Your Account has been verified by our team. You Can login using your password - " + str(str(number))
    msg.attach(MIMEText(body, 'plain'))
    s.send_message(msg)

    return '''<script>alert("accepted"); window.location='/viewhealthpending_request'</script>'''

@app.route('/healthproviderrequest_reject/<id>')
def healthproviderrequest_reject(id):
    qry = "update health_provider set status='Rejected' where hpid='" + str(id) + "'"
    ob = Db()
    res = ob.update(qry)
    return '''<script>alert("rejected"); window.location='/viewhealthpending_request'</script>'''


@app.route('/view_accepted_healthpenders')
def view_accepted_healthpenders():
    qry="select * from health_provider where status='Approved'"
    ob=Db()
    res=ob.select(qry)
    return render_template("view HP.html",data=res)

@app.route('/healthproviderrequest_rejected/<id>')
def healthproviderrequest_rejected(id):
    qry = "update health_provider set status='Rejected' where hplid='" + str(id) + "'"
    ob = Db()
    res = ob.update(qry)
    qry1 = "update login set type='pending' where lid='" + str(id) + "'"
    res1 = ob.update(qry1)
    return '''<script>alert("rejected"); window.location='/view_accepted_healthpenders'</script>'''
```

Database Design

Insert the database desgin here (if any). A sample format is given below

Attribute Name	Datatype	Width	Description
lid	Integer		Primary Key
username	Varchar	50	Unique
password	Integer		Not Null
type	Varchar	15	Not Null

Table A.1: Login

Attribute Name	Datatype	Width	Description
cid	Integer		Primary Key
complaintsender	Varchar	50	Unique
complaint	Varchar	500	Not Null
date	date		Not Null
reply	Varchar	500	Not Null
status	Varchar	30	Not Null

Table A.2: Complaint

Attribute Name	Datatype	Width	Description
did	Integer		Primary Key
pid	Integer		Unique
document	Varchar	500	Not Null
hplid	Integer		Not Null
title	Varchar	500	Not Null
date	date		Not Null
keyy	Integer		Not Null

Table A.3: Document

Attribute Name	Datatype	Width	Description
hpid	Integer		Primary Key
hpname	Varchar	100	Not Null
place	Varchar	50	Not Null
post	Varchar	50	Not Null
pin	Integer		Not Null
district	Varchar	50	Not Null
state	Varchar	50	Not Null
number	Integer		Not Null
email	Varchar	30	Not Null
hplid	Integer		Not Null
status	Varchar	15	Not Null
website	Varchar	30	Not Null
about	Varchar	500	Not Null

Table A.4: Health Provider

Attribute Name	Datatype	Width	Description
icid	Integer		Primary Key
name	Varchar	100	Not Null
place	Varchar	50	Not Null
post	Varchar	50	Not Null
pin	Integer		Not Null
district	Varchar	50	Not Null
state	Varchar	50	Not Null
number	Integer		Not Null
email	Varchar	30	Not Null
icid	Integer		Not Null
status	Varchar	15	Not Null
website	Varchar	30	Not Null
about	Varchar	500	Not Null

Table A.5: Insurance company

Attribute Name	Datatype	Width	Description
pid	Integer		Primary Key
name	Varchar	100	Not Null
place	Varchar	50	Not Null
post	Varchar	50	Not Null
pin	Integer		Not Null
district	Varchar	50	Not Null
state	Varchar	50	Not Null
number	Integer		Not Null
email	Varchar	30	Not Null
dob	date		Not Null
gender	Varchar	10	Not Null
pid	Integer		Not Null
photo	Varchar	100	Not Null
providerlid	Integer		Not Null

Table A.6: User

Attribute Name	Datatype	Width	Description
poid	Integer		Primary Key
po$_n ame$	Varchar	50	Not Null
amount	Integer		Not Null
decription	Varchar	500	Not Null
rule	Varchar	500	Not Null
insurance$_l id$	Integer		Not Null

Table A.7: Policy

Attribute Name	Datatype	Width	Description
rid	Integer		Primary Key
plid	Integer		Not Null
report	Varchar	100	Not Null
date	date		Not Null

Table A.8: Report

Attribute Name	Datatype	Width	Description
rqid	Integer		Primary Key
date	date		Not Null
policyid	Integer		Not Null
status	Varchar	15	Not Null
lid	Integer		Not Null

Table A.9: Request

DaTaflow Diagram

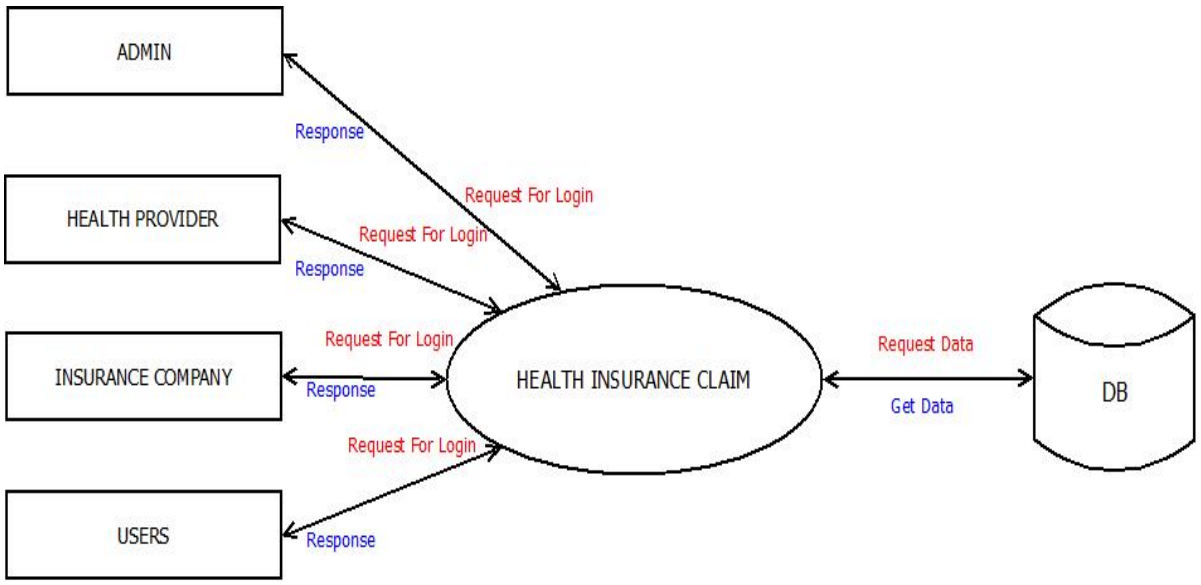

Figure A.1: Dataflow Diagram-Level-0

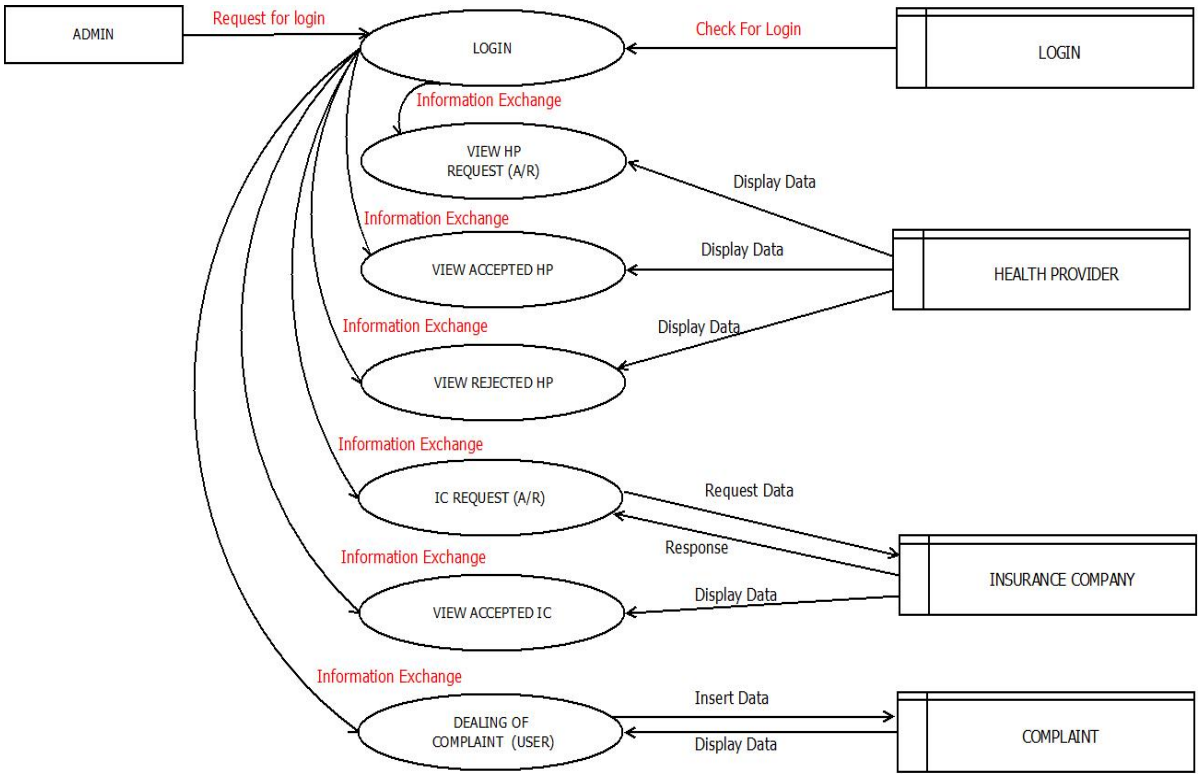

Figure A.2: Dataflow Diagram-Level-1

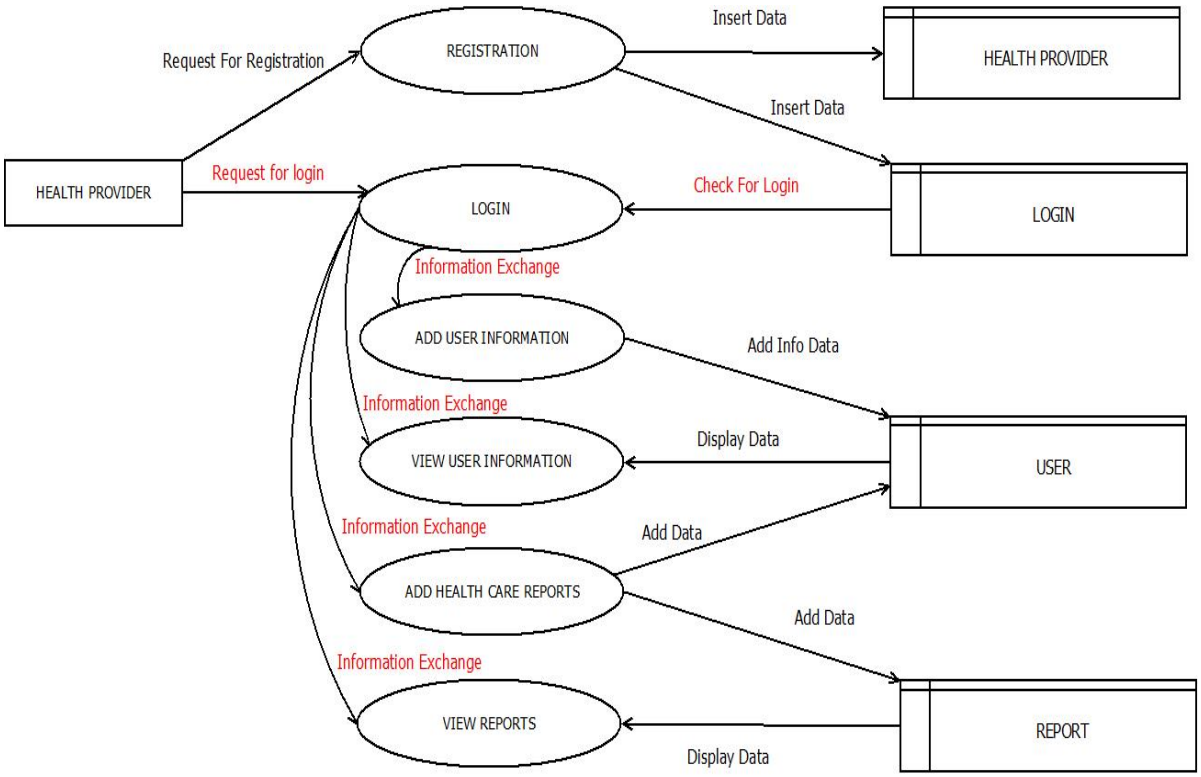

Figure A.3: Dataflow Diagram-Level-2

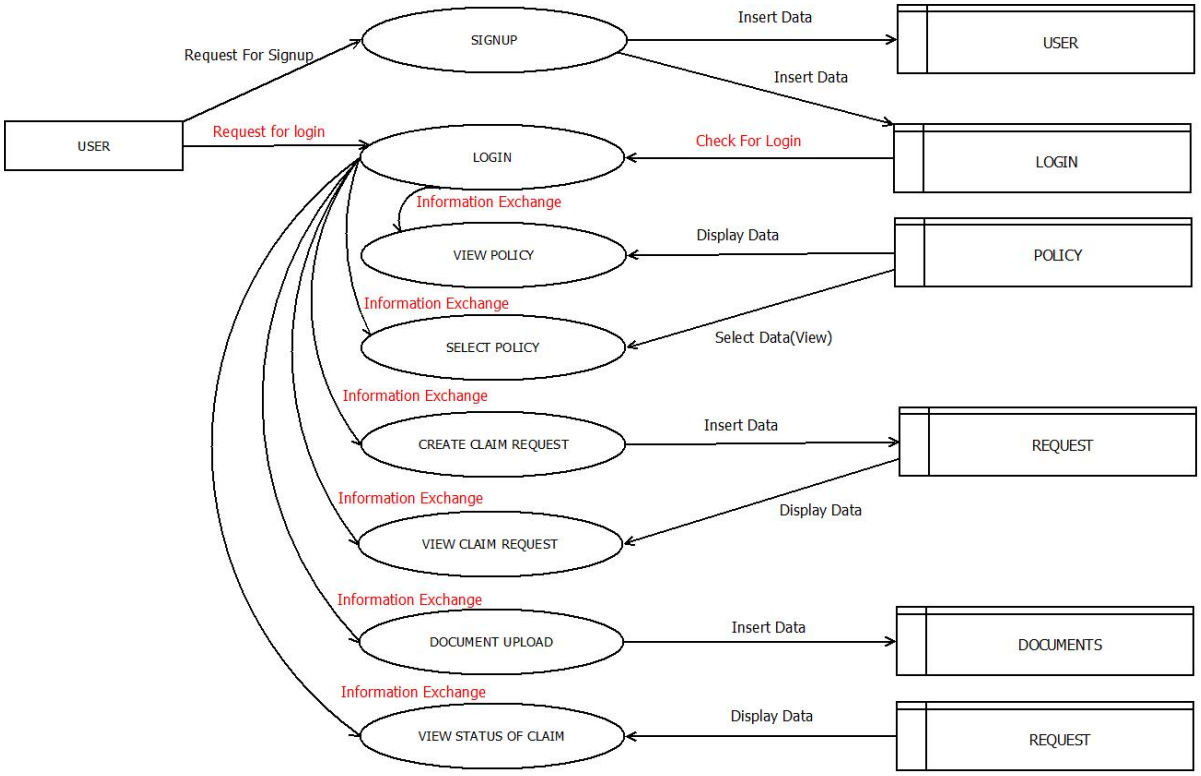

Figure A.4: Dataflow Diagram-Level-3

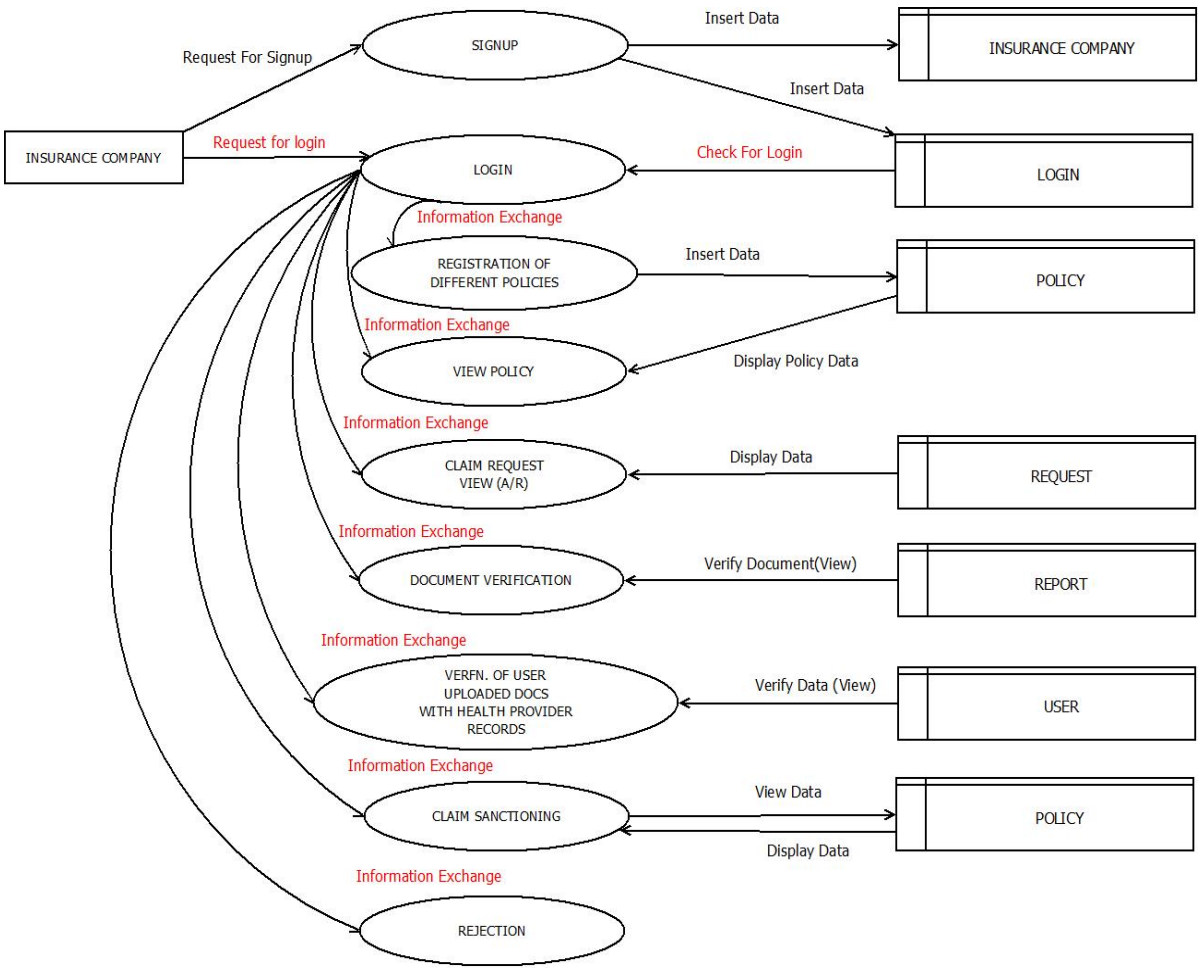

Figure A.5: Dataflow Diagram-Level-4